International Journal of Applied Engineering
Research and Development (IJAERD)
ISSN(P): 2250-1584; ISSN(E): 2278-9383
Vol. 4, Issue 2, Apr 2014, 89-96

A REVIEW ON WATERMARKING RELATIONAL DATABASES

SWATI BILAPATTE[1], SUMIT BHATTACHARYA[2] & SUDHIR SAWARKAR[3]

[1]M.E Student, Department of Computer, MGM College of Engineering and Technology, Mumbai, Maharashtra, India

[2]Department of Computer, MGM College of Engineering and Technology, Mumbai, Maharashtra, India

[3]Department of Computer, Datta Meghe College of Engineering, Mumbai, Maharashtra, India

ABSTRACT

In recent years, watermarking relational database has become a hotspot. With the rapid growth and use of internet and related technologies, especially availability of relational data over the internet, demands an effective mechanism for copyright protection with which data owner can identify the pirated copy of their data. Information hiding technique used in some host content for embedding a mark is called Watermarking. The main aim of database watermarking is to deal with the legal issue of copyright protection of database content. Many relational database watermarking techniques have been proposed to serve this purpose. This paper strongly focuses on the review of four relational database watermarking techniques proposed by researchers [R.Agarwal & Jerry Kiernam, ZHU Qin, Haiqing Wang, and Ashraf Odeh & Ali Al-Haj].

KEYWORDS: Copyright Protection, Database Watermarking, Relational Database, Watermark Embedding, Watermark Detection.

1. INTRODUCTION

Earlier, watermarking was introduced for image processing and later it extended for security of text and multimedia data such as images, video, audio sources etc. Now-a-days, it is also used for security in databases.

Watermark is secret code that is to be hidden in an imperceptible manner in the perceptive contents of the database. It is important that for every tuple of the relation, the owner of the database is responsible for identifying attribute that define the perceptive part of the relation where the watermark should be actually embedded.

Generally, there are two phases involved in Database Watermarking Techniques.

- Watermark Embedding Phase.
- Watermark Detection Phase.

Watermark embedding phase is also referred to as watermark insertion phase as a watermark is inserted into the original database. For security purpose, a secret key (private key), known only to the owner of the database is generated for inserting the watermark. The data and watermark are in-separable, after embedding the watermark. Then, the watermarked database is made publicly available. On the other hand, to verify the ownership of a suspicious database, watermark detection process (also referred as verification process) is carried out by taking the suspicious database as an input. With the help of secret key, embedded watermark is extracted and compared with the original watermark information. The watermark should be robust against tuple insertion, alteration and deletion operations performed on relational

databases. The main application of watermarking relational database is copyright protection, proof of ownership and ownership identification. Figure depicts the basic database watermark insertion and detection process.

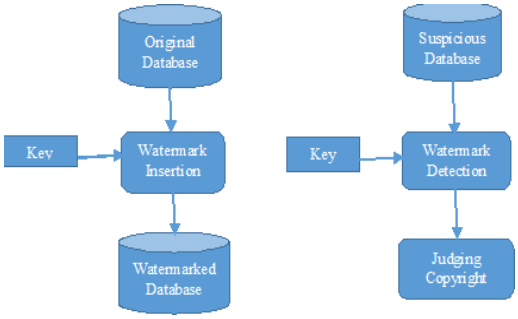

Figure 1: Database Watermark Insertion and Detection Process

According to R. Agarwal, piracy of digital assets can be protected by inserting a digital watermark into the data thus providing a promising way to protect digital data from illicit copying and manipulation [3][7]. Tremendous growth in digital watermarking has increased interest in intellectual property and copyright protection [7][11].

The organization of paper is as follows: section 2 describes the literature survey done by different researchers. Section 3 elaborates on comparative analysis of different database watermarking techniques. Section 4 concludes the paper and suggests future directions.

2. LITERATURE SURVEY

With the growing use of relational databases, especially the expanded availability and use of digital assets such as software, images, video, audio, text etc. over the internet, security of such relational data has been a great concern. Hence, digital watermarking for relational databases emerged as a candidate solution to provide copyright protection, tamper detection, traitor tracing and maintaining integrity of relational data [10].

The desiderata for watermarking relational databases are:

Robustness: Robust means (durable to changes) that embedded watermark should be able to survive under benign and unintentional attacks.

Accuracy: The owner of database should not detect his/her watermark in someone else's non pirated database.

Capacity: It is the maximum amount of data that can be embedded.

Detectability: The owner of the database should be able to detect the watermark by examining the tuples from the suspicious database [7].

Incremental Updatability: Watermark should be incrementally updatable such that tuples of the relational databases are either added/deleted or modified, the watermark value should not be changed.

Blind System: Watermark detection process should not require the knowledge of either the original database or the watermark.

Public System: According to Kirchhoff's, method used for inserting the watermark should be public. Defence should lie only in the selection of private (secret) key.

Challenges for watermarking relational databases

The watermarked database may suffer from both deliberate and unintentional attacks, in which a watermark can potentially be damaged erased or compromised as described below:

- **Benign Updates:** Tuples of any watermarked relation are processed as usual. As a result, the marked tuples may be added, deleted or updated which may remove or cause the embedded watermark undetectable [8].
- **Malicious Attacks**: Attacker may steal the data and try to erase the watermark [3].

Malicious attacks may take various forms:

- **Bit Attacks:** Malicious attacker attempts to destroy the watermark by altering/updating one or more bits.
- **Bit Flipping Attacks:** It inverts the values of some of the number of bit positions [2] [7].
- **Randomization Attacks:** It assigns random values to some number of bit positions [2] [7].
- **Rounding Attacks:** Attacker may try to lose the marks contained in a numeric attribute by rounding all the values of the attribute [3].
- **Transformations:** here, numeric values are linearly transformed. For example, Attacker may convert the data to different unit of measurement (e.g., Fahrenheit to Celsius) [3].
- **Subset Attacks:** Attacker may take a subset of the attributes (or tuples) of a watermarked relation hoping that the watermark is lost.
- **Mix-and-Match Attacks**: Attacker creates his/her own relation by considering disjoint tuples from multiple relations that contains similar information.
- **Insertion Attack:** Here, attacker inserts tuples to the data set hoping to disturb the embedded watermark which results in synchronization errors [9].
- **Alteration Attack:** Attacker alters the data values of the tuples which lead to disturbance in the watermark. Altering the data values violates the usability constraints making the data useless.
- **Deletion Attack:** Attacker tries to delete the marked tuples from the relational database leading to synchronization errors.

3. COMPARATIVE ANALYSIS

A. Algorithm Proposed by R. Agarwal & J. Kiernan for Watermark Insertion and Detection [3]

Agarwal & Kiernan, proposed a watermarking technique that marks the only numeric attribute of a relation assuming that the marked attribute can tolerate small changes in some of their values without affecting the quality of data. All of the numeric attribute of a relation need not to be marked. However, in this case data owner is responsible for deciding which attributes are suitable for marking. This algorithm is based on primary key of a relation and secret (private) key that inserts the watermark bits in the least significant bits (LSB) of selected attributes of a selected subset of tuples of a relation. Only if one has access to the secret key, then there is a high probability of detecting the watermark.

The following figure summarizes the important parameters used in the algorithm.

Ƞ	Total number of tuples in the relation
υ	Number of attributes available for marking in the relation
Z	Number of least significant bits in an attribute available for marking
1/ϒ	Target fraction of marked tuples
Ꞷ	Actual number of marked tuples
A	Significance level of the test for watermark detection
Ƭ	Threshold parameter for watermark detection

Figure 2: Notation

// The secret key κ is known only to the database owner, g is a gap parameter.

// The parameters r (relation), υ, ζ, i (attribute index), j (bit index) are also private to the data owner.

1. for each tuple r ϵ R do
2. if (F(r●P) % g) == 0) then // mark this tuple
3. attribute_index i = F(r●P)mod υ // mark attribute A_i
4. bit_index j = F(r●P)mod ζ // mark j^{th} bit
5. r●A_i = mark(r●P, r●$A_{i,}$ j)
6. mark the primary key k & bit index j
7. calculate the hash value of secret key and primary key κ of the database using Message Authentication Code (MAC)
8. if (hash_value == even) then set LSB bit of the attribute to 0 else set to 1
9. return υ

Figure 3: Watermark Insertion Algorithm

// parameters considerations are same as followed by the watermark insertion algorithm.

// α is the test significance level that the detector preselects.

1. Total_count = match_count = 0
2. For each tuple s ϵ S do
3. if (F(s●P) % g) == 0) then // tuple was marked
4. attribute_index i = F(s●P)mod υ // A_i was marked
5. bit_index j = F(s●P) mod ζ // j^{th} bit was marked
6. Total_count = Total_count + 1 // increment the Total_count by 1
7. Match_count = Match_count + Match(s●P, s●$A_{i,}$ j)
8. Ƭ = threshold (Total_count, α) // Calculate the threshold
9. if (Match_count ≥ threshold) then suspect piracy
10. match the secret key, attribute_index and bit_index of relation R & return integer
11. calculate the hash value of secret key and primary key κ of the database using Message Authentication Code (MAC)
12. if (hash_value = even) then return 1 if set LSB bit of the attribute is 0 else return 0
13. else return 1 if LSB bit of the attribute is 1 else return 0

Figure 4: Watermark Detection Algorithm

This technique can't be used for multi-bit watermark and is also not resilient to insertion, alteration, and deletion attacks.

B. Algorithm Proposed by ZHU Qin, YANG Ying, LE Jia-jin, LUO Yi-shu for Watermark Embedding and Detection [4]

Zhu Qin proposed a novel scheme based on digital watermarking for protecting the copyright of outsourced database. Method proposed by these authors is almost same as the method proposed by R. Agarwal & J. Kiernan only difference is about primary key attribute.

In the first approach primary key along with the secret key was used to generate the watermark whereas, in this approach watermark is generated from chaotic random number which is generated from both the primary key & secret key. Thus under the control of key and chaotic random number watermark is embedded into the database. No original database is needed during watermark detection, and the copyright is judged by the match rate of watermark. The capacity of the watermark is limited in previous approach and the scheme is not suitable for database which needs frequent updating, as it is very costly to re-embed watermark into the updated database.

The scheme of a database relation is R (P, A_0, . . . , Av-$_1$), where P is the primary key attribute

The steps of watermark embedding are outlined as follows:

1. Input the value of gap γ, is the interval number between two adjacent marked tuples;
2. According to data range and the precision of A_j for embedding, compute the value of ξj, which determine the number of bits of LSB of A_j.
3. For each tuple r_i, repeat:

   3.1 Compute the Bits Conjunction of the Secret Key and the Primary key ($K \bullet P_i$) and Normalize it, s.t. $0<NRM(K \bullet P_i) < 1$;

   3.2 if LGS (NRM ($K \bullet P_i$)) mod $\gamma = 0$, then

   3.2.1 j = (NXT (LGS (NRM ($K \bullet P_i$))) mod v) + 1;

   3.2.2 κ = NXT (LGS (NRM ($K \bullet P_i$))) mod ξj;

   3.2.3 MRK_{ij} = (NXT (LGS (NRM ($K \bullet P$))) mod 2)? 1:0;

   3.2.4 $r_i \bullet A_j$ = WTR ($r_i \bullet A_j$, MRK_{ij}, κ)

Figure 5: Watermarking Embedding Algorithm

Watermark detection is the reverse process of the embedment; its steps are similar to the embedment algorithm.

- For each tuple do
- Compute the mark value, determined by the algorithm of watermark embedment.
- Compare the computed value with the read value
- Count the matched bits. CNT denotes the total matched bits.
- The count of all the marked bits of the database that has not suffered from watermark attack should be n/ γ, so the watermark match rate equals CNT/ (n/ γ).

Figure 6: Watermark Detection Algorithm

C. Algorithm Proposed by Haiquig Wang, Xinachin Cui and Zaihi Cao for Watermark Generation and Detection [5]

Haiquing, proposed a speech based algorithm for watermarking relational database. Same as the previous approaches but here speech and secret message has been used as watermark. Speech is suitable for watermarking due to its distinguishable characteristics such as measurable, uniqueness, stability, and universality.

Watermark Generation Algorithm

- Compress the speech signal using wavelet transform.
- Speech signal enhancement should be done in two parts: Estimate the spectrum of the background noise. Subtract the noise spectrum from speech.
- Convert speech signal waveform to the 8-bit A-law.
- Generate the watermark using the message of the copyright holder.

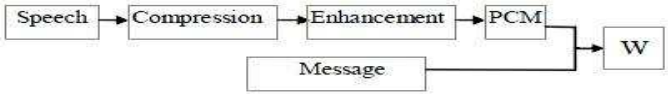

Figure 7: Watermark Generation Algorithm [5]

D. Algorithm Proposed by Ashraf Odeh & Ali Al-Haj for Watermark Embedding and Extraction [6]

Ashraf Odeh has proposed an approach that hides the watermark information (bits) in the unnoticed 'time' attribute of the selected tuple of a relation in the database. Binary image has been used as watermark. In database DATE field consist of two parts: 'Date' & 'Time', here (SS) i.e. seconds field is used from the time field (HH:MM:SS) to hide the binary information of watermark. One of the major advantages of using the time attribute is the large bit-capacity available for hiding the watermark, thus large watermark can easily be hidden, if required.

Watermark Embedding

1. Transfer the image into a flow of bits.
2. Group every 5 bits as a binary string and find the decimal equivalent of the string.
3. Embed the decimal number in tuples selected by the pre-defined key 'K' as follows:

3.1 For each selected tuple do

3.2 For each selected 'time' attribute do

3.3 if the 'SS' field of the 'time' mode K=0

3.4 Embed the decimal number

3.5 Else Next attribute

3.6 End if

3.7 End loop

3.8 End loop

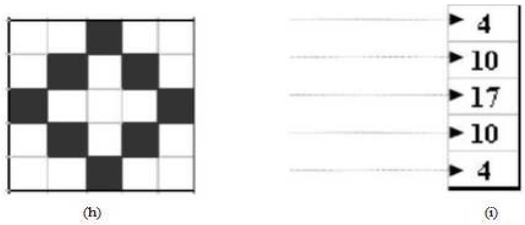

Figure 8: Binary Image Watermark and i) Its Decimal Equivalent Vector [6]

Watermark extraction is the reverse process of the watermark embedding.

- Extract the decimal number in tuples selected by the pre-defined key 'K' procedure.
- Find the binary equivalent of the extracted decimal number.
- Group every 5 bits as a binary string and reconstruct the binary image watermark from the binary strings.

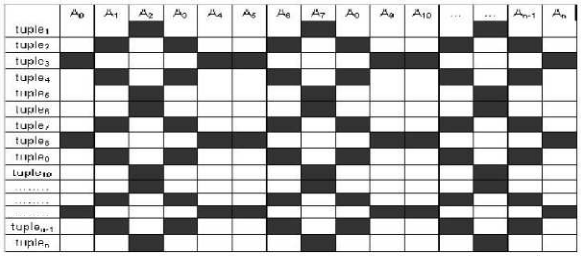

Figure 9: Snapshot of Watermarked Database [6]

Table summarizes the comparison between above described watermarking techniques with respect to characteristics and different types of attacks.

Table 1: Comparative Analysis of Different Database Watermarking Techniques

Characteristics	Proposed Schemes			
	R. Agarwal & J. Kiernan (2002)	ZHU Qin (2006)	Haiquig Wang (2008)	Ashraf Odeh & Ali Al-Haj (2008)
Watermark Information	Bit pattern	Bit pattern	Database content	Database content
Cover type	Numeric	Numeric	Categorial	Numeric
Granularity type	Bit level	Bit level	Attribute value	Bit level
Intent	Proof of Ownership	Proof of Ownership	Proof of Ownership	Proof of ownership
Irrepressible to subset selection attack	No	No	Yes	Yes
Irrepressible to subset insertion attack	No	Yes	Yes	No
Irrepressible to subset alteration attack	No	Yes	Yes	Yes
Irrepressible to subset deletion attack	No	No	Yes	Yes

4. CONCLUSIONS

In this paper, we reviewed four papers proposed by different authors on watermarking relational databases. In each of the paper, author embeds a watermark bit into the database so as to protect the data from various types of intentional and unintentional attacks. Every author worked for the robustness of the technique and claims proof of ownership. R. Agarwal [3] and ZHU Qin [4] used watermarking for only numeric data. Haiquing wang [5] has applied the

watermarking scheme for numerical and categorial data as well. Algorithm proposed by Haiquig Wang [5] is more secure and robust because speech which has distinctive features is used as a watermark.

REFERENCES

1. F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on Copyright Marking Systems", Lecture Notes in Computer Science, 1525:218– 238, April 1998.
2. R. Agrawal and J. Kiernan, "Watermarking relational databases", In Proceedings of The 28th International Conference on Very Large Databases VLDB, 2002.
3. R.Agrawal, P. J. Hass, J. Kiernan, "Watermarking relational data: Framework, Algorithms and Analysis", VLDB Journal, 2003, pp.157-169.
4. ZHU Qin, YANG Ying, LE Jia-jin, LUO Yishu (2006), "Watermark based Copyright Protection of Outsourced Database," IEEE, IDEAS, pp. 1-5.
5. Haiquig Wang, Xinachin Cui, and Zaihi Cao, "A speech based Algorithm for Watermarking Relational Database", page no 603-606, IEEE, ISIP-2008
6. A. Odeh and A. Al-Haj "Watermarking Relational Database Systems", IEEE, pp. 270-274, 2008.
7. Prof.Bhawana Ahire, Prof. Neeta Deshpande," Watermarking relational databases: A Review", IOSR Journal of Engineering (IOSRJEN) ISSN: 2250-3021 Volume 2, Issue 8.
8. Mayuree K.Rathva, Prof.G.J.Sahani, "WATERMARKING RELATIONAL DATABASES", International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.1, February 2013
9. Mohamed Shehab, Elisa Bertino and Arif Ghafoor, "Watermarking Relational Databases Using Optimization-Based Techniques", IEEE Transaction on Knowledge and Data engineering, VOL. 20, NO. 1, JANUARY 2008.
10. Raju Halder, Shantanu Pal, Agostino Cortesi (2010), "Watermarking Techniques for Relational Databases: Survey, Classification and Comparison" Journal of Universal Computer Science, vol. 16, no. 21, 3164-3190.
11. I. Cox, J.Bloom, and M.Miller, "Digital Watermarking", Morgan Kaufmann, 2001.
12. R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data", IEEE Transactions on Knowledge and Data Engineering, 16(6), June 2004.
13. Ali Al-Haj and Ashraf Odeh, "Robust and Blind Watermarking of Relational Database Systems**",** Journal of Computer Science 4 (12): 1024-1029, 2008 ISSN 1549-3636© 2008 Science Publications
14. M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", Proceedings of the IEEE, 86:1064–1087, June 1998.